Scribing-0803_second class

Liuxuan Yu, Mengnan He, Anying Li

August 3, 2016

K-means is fast and efficient, be the first try. The second go-to algorithm: hierarchical clustering.

Strength of Hierarchical clustering:

(1) Do not have to decide the number of cluster upfront
(2) Can be very interpretable.

Limitations of Hierarchical clustering:

(1) Computational complexity: O(N2)
  – as the size of data increase by 10 times, the computational time will increase by 100 times
  – the complexity can be O(N3) in the worst case (as the size of data increase by 10 times, the computational time will increase by 1000 times)

(2) Choice of linkage is important – sensitive to noise/outliers – difficulty of different sized clusters

(3) Once you decided to break or combine a cluster, you can't undo it.

(4) No objective function is directly minimized

Two categories of Hierarchical clustering:

(1) Divisive: from trunk and then upside down

(2) Agglomerative: from leaf to the whole tree
  – start everything with its own cluster;
  – collection of points and clusters

Key input: linkage criterion

How to split and merge? how to define the distance?

1. Centroid: distance between centers
2. Min/single: distance between the closest points/clusters
3. Max/complete: distance between furthest points or clusters
4. Average: average distance of all possible pairs

- Whether you choose divisive or agglomerative, they both need linkage criterion!

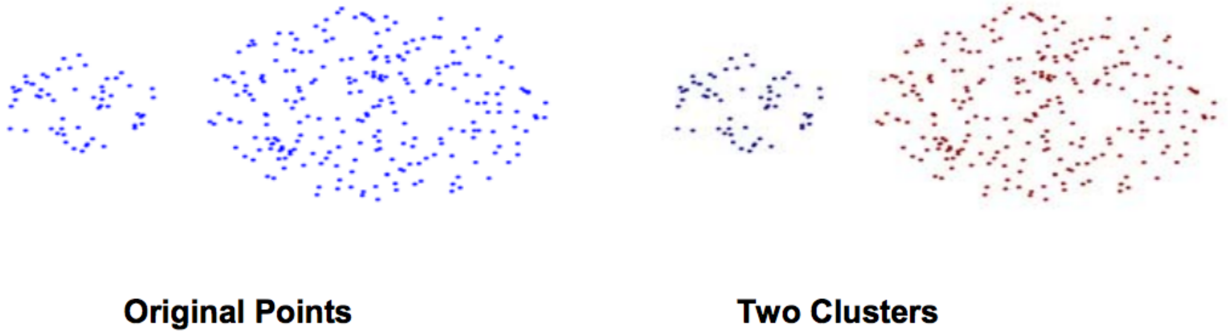


Figure 1: **Strength of min/single linkage: unlike some other clustering methods (e.g. K-Means), hierarchical clustering with min/single linkage can handle non-elliptical shapes.**

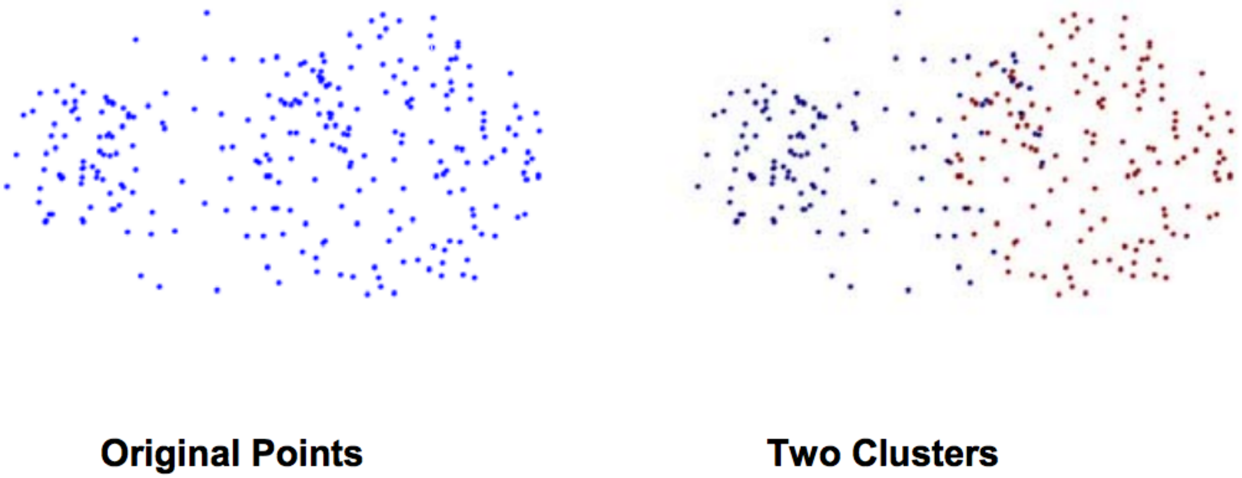
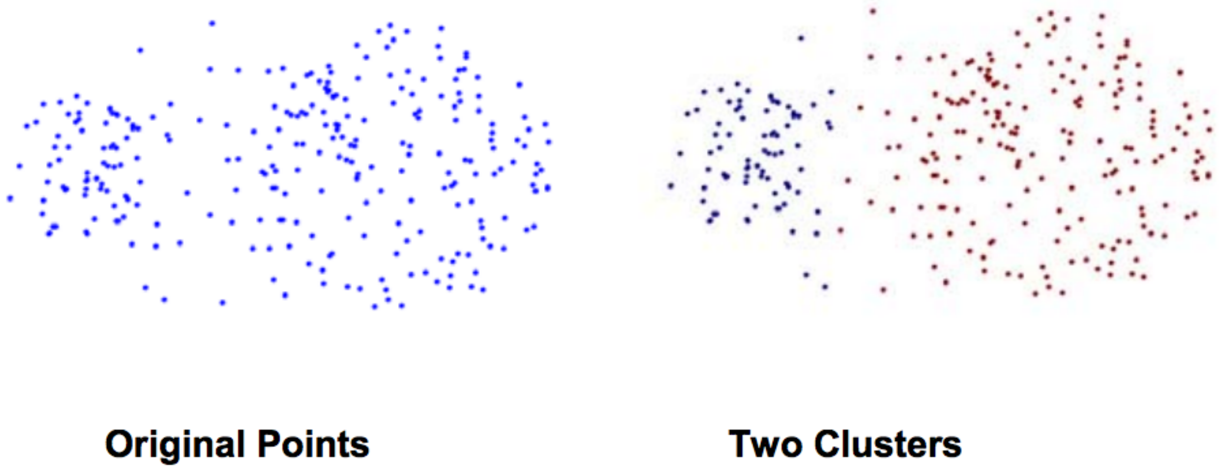


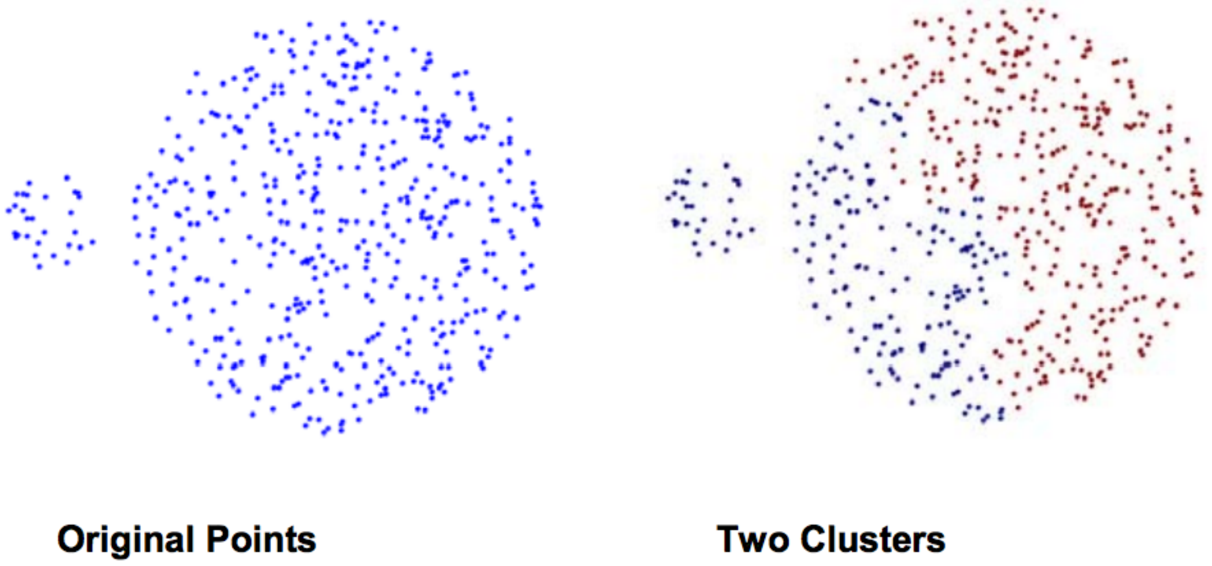
Figure 2: **Weakness of min/single linkage: this measurement of distance is very sensitive to outliers. Simply adding some meaningless noise around the original shapes in Figure 1 will cause confusion to the clustering result.**



Original Points

Two Clusters

Figure 3: **Strength of max/complete: more robust to noise and outliers. From Figure 3 we can tell that max/complete linkage sets a much clearer boundary between the two clusters than the min/single linkage in Figure 2.**



Original Points

Two Clusters

Figure 4: **Weakness of max/complete: similar to the K-Means method, hierarchical clustering with max/complete linkage tends to break large clusters.**

the differences between using single and complete linkage in hierarchical clustering for the protein-consumption data set

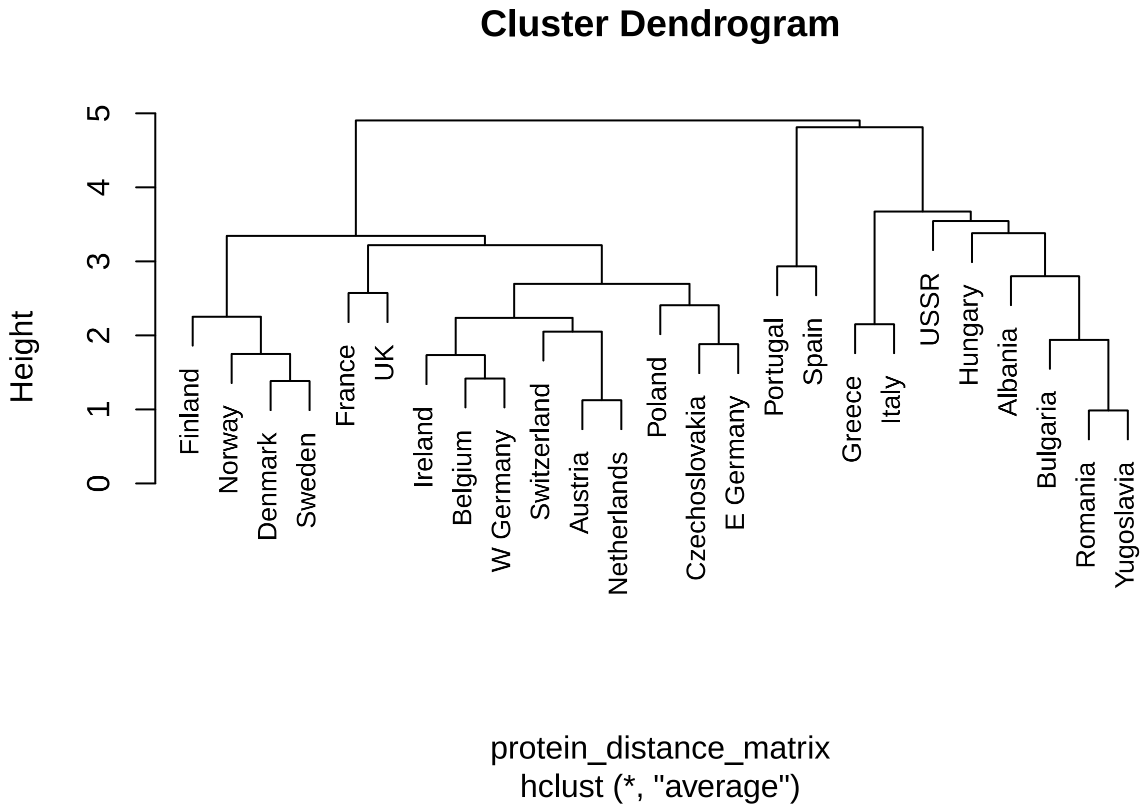
```
protein <- read.csv("protein.csv", row.names=1)
protein_scaled <- scale(protein, center=TRUE, scale=TRUE)
```

1. Form a pairwise distance matrix using the dist function

```
protein_distance_matrix = dist(protein_scaled, method='euclidean')
```

2. Run hierarchical clustering and then cut the tree into 5 clusters.

```
hier_protein = hclust(protein_distance_matrix, method='average')
plot(hier_protein, cex=0.8)
```



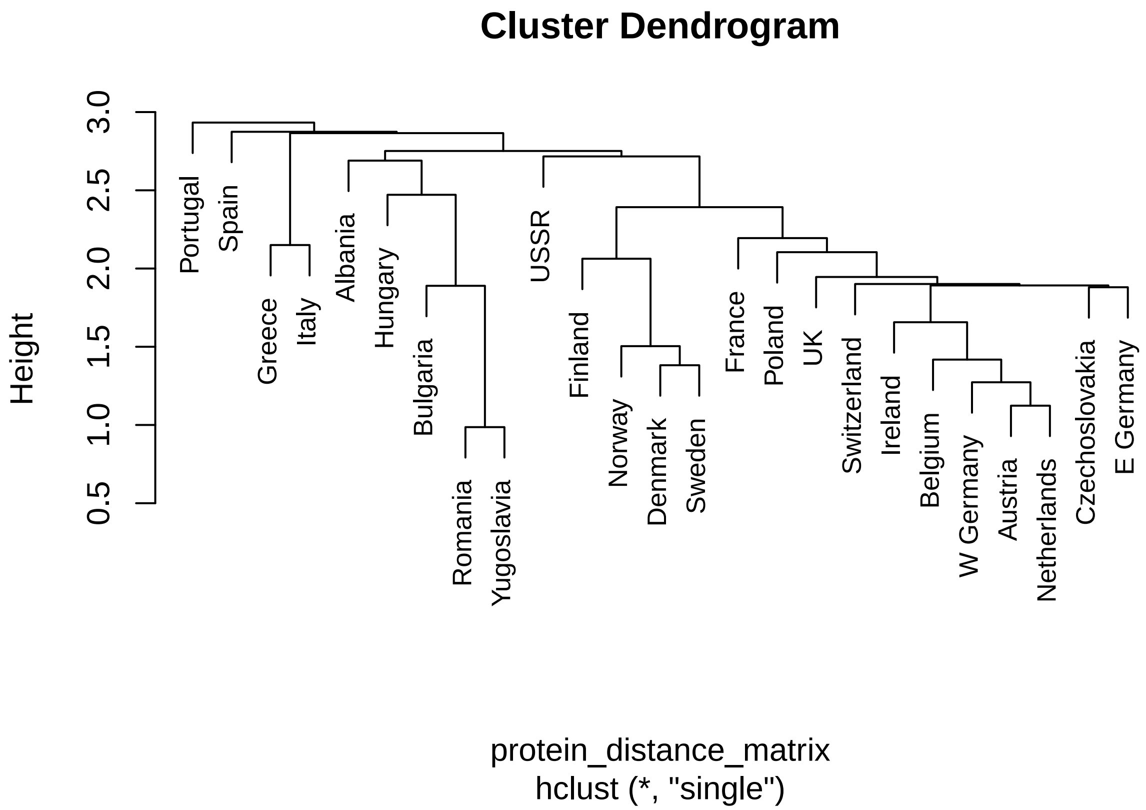
```
cluster1 = cutree(hier_protein, k=5)
```

3. This is to tell which cluster each country is in

```
which(cluster1 == 1)
which(cluster1 == 2)
which(cluster1 == 3)
```

4. Use single linkage instead. The structure of the tree is more unbalanced than the first one.

```
hier_protein2 = hclust(protein_distance_matrix, method='single')
plot(hier_protein2, cex=0.8)
```


Cluster Dendrogram
Height
Portugal
Spain
Greece
Italy
Albania
Hungary
Bulgaria
Romania
Yugoslavia
USSR
Finland
Norway
Denmark
Sweden
France
Poland
UK
Switzerland
Ireland
Belgium
W Germany
Austria
Netherlands
Czechoslovakia
E Germany
protein_distance_matrix
hclust (*, "single")